

AMENDMENTS TO THE CLAIMS

Please amend the claims of the present application as set forth below.

Claims 1-38 were originally filed.

No claims have been cancelled.

No new claims have been added

Accordingly, claims 1 – 38 are pending.

AI

1. (Currently Amended) A method comprising:
loading a reflection image into memory;
retrieving an environment texture sample from an environment map based
on a reflection vector stored in a pixel of the reflection image; and
applying the environment texture sample to ~~the~~ an object.

2. (Original) The method of claim 1, wherein the loading comprises:
retrieving a reflection texture sample comprising red, green, and blue color
data; and
storing the red, green, and blue color data of the reflection texture sample as
red, green, and blue color data of a pixel of the object.

3. (Original) The method of claim 2, wherein the retrieving comprises
interpreting the red, green, and blue color data of the pixel as the reflection vector.

1 4. (Original) The method of claim 3, wherein the retrieving comprises
2 retrieving the environment texture sample comprising red, green, and blue color
3 data from the environment map based on the interpreted reflection vector.

4
5 5. (Original) The method of claim 4, wherein the applying comprises
6 replacing the red, green, and blue color data of the pixel with the red, green, and
7 blue color data of the environment texture sample.

8
9 6. (Original) The method of claim 3, further comprising perturbing the
10 interpreted reflection vector prior to retrieving the environment texture sample.

11
12 7. (Original) The method of claim 1, wherein the loading, the
13 retrieving, and the applying are performed during a single pass through a graphics
14 pipeline.

15
16 8. (Original) The method of claim 6, further comprising storing a result
17 in a frame buffer.

18
19 9. (Original) The method of claim 1, wherein the loading is performed
20 during a first pass through a graphics pipeline and the retrieving and the applying
21 are performed during a second pass through the graphics pipeline.

22
23 10. (Original) The method of claim 9, further comprising:
24 storing the reflection image in a frame buffer; and
25

1 replacing the reflection image in the frame buffer with a result of
2 application of the environment texture sample.

3
4 11. (Original) The method of claim 10, further comprising:
5 loading the reflection image in a texture memory; and
6 loading the environment map in the texture memory prior to performing the
7 retrieving and the applying.

8
9 12. (Original) The method of claim 1, wherein the retrieving comprises
10 retrieving the environment texture sample from a cube environment map.

11
12 13. (Original) The method of claim 1, further comprising:
13 generating a plurality of reflection images, wherein each of the plurality of
14 reflection images corresponds to a particular viewpoint; and
15 loading a predetermined reflection image chosen from the plurality of
16 reflection images into the memory.

17
18 14. (Original) The method of claim 1, wherein the loading, the
19 retrieving, and the applying are performed in real time.

20
21 15. (Original) A method comprising:
22 retrieving a texture sample from a texture map, the texture sample
23 containing reflection data;
24 using the reflection data in the texture sample to obtain an environment
25 texture sample in an environment map; and

1 applying the environment texture sample to an object.

2
3 16. (Original) The method of claim 15, wherein the applying comprises
4 replacing red, green, and blue color data of the pixel with red, green, and blue
5 color data of the environment texture sample.

6
7 17. (Original) The method of claim 15, further comprising perturbing
8 the reflection data prior to using the reflection data to obtain the environment
9 texture sample.

10
11 18. (Original) The method of claim 15, wherein the using comprises
12 using the reflection data to obtain an environment texture sample in a cube
13 environment map.

14
15 19. (Original) The method of claim 15, further comprising storing, in
16 memory, a result of said applying the environment texture sample to the object.

17
18 20. (Currently Amended) A method comprising:
19 interpreting data in a pixel of a reflection image as a reflection vector; and
20 using the reflection vector to obtain an environment texture sample in an
21 environment map.

22
23 21. (Original) The method of claim 20, wherein the interpreting
24 comprises interpreting red, green, and blue color data of the pixel as the reflection
25 vector.

1
2 22. (Original) The method of claim 20, further comprising perturbing
3 the reflection vector prior to using the reflection vector to obtain the environment
4 texture sample.

5
6 23. (Currently Amended) The method of claim 20, wherein the using
7 comprises using the reflection ~~data~~ vector to obtain an environment texture sample
8 in a cube environment map.

9
10 24. (Original) The method of claim 20, further comprising applying the
11 environment texture sample to an object.

12
13 25. (Original) A method comprising:
14 retrieving a texture sample from a texture map; and
15 using the texture sample to retrieve an environment texture sample from an
16 environment map.

17
18 26. (Original) A computer program product comprising a computer
19 useable medium having computer program logic recorded thereon for enabling a
20 processor to render a computer scene, the computer program logic comprising:

21 a texture map comprising reflection data;
22 a texture map sampling procedure that enables the processor to obtain a
23 first texture sample from the texture map and apply the first texture sample to an
24 object;
25 an environment map; and

1 an environment map sampling procedure that enables the processor to
2 obtain a second texture sample from the environment map based on the first
3 texture sample and apply the second texture sample to the object.
4

5 27. (Original) The computer program product of claim 26, wherein the
6 texture map sampling procedure enables the processor to obtain red, green, and
7 blue color data from the texture map and store the red, green, and blue color data
8 as a pixel of the object.
9

10 28. (Original) The computer program product of claim 27, wherein the
11 environment map sampling procedure enables the processor to use the red, green,
12 and blue color data of the pixel as a reflection vector to obtain the second texture
13 sample.
14

15 29. (Original) The computer program product of claim 26, wherein the
16 environment map comprises a cube environment map.
17

18 30. (Original) The computer program product of claim 26, further
19 comprising a texture map generating procedure that enables the processor to
20 generate a particular texture map comprising reflection data based on a particular
21 viewpoint.
22

23 31. (Original) A computer readable medium storing computer-readable
24 instructions that, when executed, direct a graphics processing system to:
25

1 interpret data in one of a reflection image or a texture map as a reflection
2 vector;

3 use the reflection vector to obtain an environment texture sample in an
4 environment map; and

5 apply the environment texture sample to an object.

6
7 32. (Original) The computer readable medium of claim 31, further
8 comprising computer-readable instructions that, when executed, direct the
9 graphics processing system to interpret red, green, and blue color data as the
10 reflection vector.

11
12 33. (Original) The computer readable medium of claim 31, further
13 comprising computer-readable instructions that, when executed, direct the
14 graphics processing system to perturb the reflection vector prior to using the
15 reflection vector to obtain the environment texture sample.

16
17 34. (Original) A system for environment mapping, comprising:
18 an application program having computer program logic, an environment
19 map, and a texture map comprising reflection data;
20 a processor for implementing the computer program logic; and
21 a graphics subsystem, under the control of the application program and the
22 processor, for rendering an object using the texture map and the environmental
23 map.

1 35. (Original) The system of claim 34, wherein the graphics subsystem
2 comprises:

3 a texture memory;

4 a frame buffer; and

5 a rasterizer coupled to the texture memory and the frame buffer.

6
7 36. (Original) The system of claim 34, wherein the graphics subsystem
8 is configured to use a texture sample retrieved from the texture map to obtain an
9 environment texture sample from the environment map and to apply the
10 environment texture sample to the object.

11
12 37. (Original) The system of claim 34, wherein the graphics subsystem
13 is configured to retrieve an environment texture sample from the environment map
14 based on a reflection vector stored in a texture sample from the texture map and to
15 apply the environment texture sample to the object.

16
17 38. (Original) The system of claim 34, wherein the graphics subsystem
18 is configured to retrieve a texture sample from the texture map, the texture sample
19 containing reflection data, the graphics subsystem being further configured to use
20 the reflection data in the texture sample to obtain an environment texture sample
21 in the environment map.